REMARKS

Claims 13, 18, 21-43, 46, 47, 52, 54, and 56 have been amended. Claims 44-45 have been canceled. Claims 1-43 and 46-59 are pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(a) Rejection:

The Examiner rejected claims 1-15, 17-32, 34-49 and 51-59 under 35 U.S.C. § 102(a) as being anticipated by Daynes (U.S. Patent 6,182,186). Applicants traverse the rejection for at least the following reasons.

Applicants note that throughout the Office Action, the Examiner refers to the Dice patent cited in the previous Office Action in his remarks, rather than the Daynes patent. Applicants assume the Examiner's citations refer to the Daynes patent. Clarification is requested.

Regarding claim 1, contrary to the Examiner's assertion, Daynes does not disclose *a method of providing non-blocking multi-target transactions in a computer system*. The Examiner cites FIG. 7 and related text in his remarks. However, nothing in Daynes describes non-blocking multi-target transactions. Instead, Daynes describes the use of a respective lock state to control access to each of a plurality of resources. FIG. 7 of Daynes illustrates the process of acquiring a lock for a single resource. This process includes placing a lock request in a queue until the lock becomes available (e.g., until it is released by another transaction), which is clearly a blocking mechanism, as would be understood by those of ordinary skill in the art. Daynes describes a non-blocking synchronization for changing the lock state associated with a single one of these resources (see, e.g., column 17, lines 2 – column 18, line 13). However, this clearly does not provide a non-blocking multi-target transaction, since this mechanism is used to access a single target, the lock state associated with a single shared resource.

Further regarding claim 1, Daynes does not disclose *defining plural transactionable locations, wherein individual ones of the transactionable locations encode respective values* *<u>and are owned by no more than one transaction at any given point in a multithreaded computation</u>*. The Examiner cites FIGs. 4 and 11 (specifically, element 1148) and column 18, lines 15-25 as teaching these limitations. The passage cited by the Examiner describes that one type of lock that a transaction may own is an exclusive lock (i.e., a single-write-owner, or SWO, lock). However, Daynes clearly describes other lock states (e.g., MRO: multiple read owner, and MWO: multiple write owner) that represent <u>ownership by multiple owners for a given resource</u>. FIG. 4, also cited by the Examiner, actually illustrates this multi-owner concept and, thus, **<u>teaches away</u> from the above-referenced limitation of claim 1**. In FIG. 4, element 408 represents a lock state with an empty write set and <u>a read owner set made up of two transactions</u>, T_1 and T_2. Element 1148 of FIG. 11, also cited by the Examiner, also supports this multi-owner functionality. This element represents a global variable comprising a set of transactions that may be ignored by other transactions in cases that would otherwise result in an ownership conflict (e.g., allowing multiple owners to be included in a read or write set, but causing a terminated transaction that has not yet been removed from the read or write set to be ignored).

Daynes also fails to disclose *wherein the ownership acquiring wrests ownership from another transaction, if any, that owns the targeted transactionable location*. The Examiner cites column 18, lines 45-50 as teaching this limitation. This passage describes the memory usage of the lock states of Daynes invention and has nothing to do with the above-referenced limitation of claim 1. Applicants assert that Daynes clearly does not teach that one transaction wrests ownership from another transaction that owns a targeted transactionable location, as recited in the claim. Instead, as illustrated in FIG. 7 and FIG. 13, a lock may not be acquired by one transaction if there is a conflict with another transaction (e.g., if another transaction holds an exclusive lock such as an SRO or SWO lock). In this case, a new lock request is placed in a queue until the conflict is resolved (e.g., the owning transaction releases the lock) and until any lock requests already pending in the queue are serviced ahead of the new lock request. In the case that a multi-

owner lock state is associated with a resource, an additional lock may be acquired. However, in this case, ownership is not wrested from any other owners. **Since in neither of these cases, nor any other scenario described in Daynes, a transaction <u>wrests ownership from another transaction</u>, Daynes clearly does not disclose the above-referenced limitation.**

Finally, Daynes fails to disclose *attempting to commit the particular multi-target transaction using a single-target synchronization primitive to ensure that, at the commit, the particular multi-target transaction continues to own each of the targeted transactionable locations*. The Examiner cites column 19, lines 1-15 and column 20, lines 5-15 in teaching these limitations. The Examiner's citation in column 19 describes that inactive bit numbers may be recycled (e.g., used in a subsequent owner set). This has absolutely nothing to do with the above-referenced limitation of claim 1. The Examiner's citation in column 20 describes how lock states may be cached and the cached lock states may be used for acquiring the lock of an unlocked resource. It also has nothing to do with the above-referenced limitation of claim 1. In addition, as discussed above, the only single-target synchronization primitive described in Daynes is used to attempt to update the lock state associated with a given resource, not to commit a multi-target transaction. Applicants assert that nothing in Daynes discloses the use of a single-target synchronization primitive in an attempt to commit a multi-target transaction, as required by claim 1.

For at least the reasons above, Daynes cannot be said to anticipate claim 1 and removal of the rejection thereof is respectfully requested.

Independent claims 22, 46, and 56 include limitations similar to those discussed above regarding claim 1. Therefore, the arguments presented above apply with equal force to these claims, as well.

Applicants also note that the Examiner has not included any remarks directed to Independent claim 46 in the Office Action, although claim 46 is listed in the rejection

under 35 U.S.C. § 102(a). Therefore, no *prima facie* rejection has been stated for claim 46.

<u>Section 103(a) Rejection</u>:

The Examiner rejected claims 16, 33 and 50 under 35 U.S.C. § 103(a) as being unpatentable over Daynes in view of Mages, et al. ("Non-Blocking Algorithms and Preemption-Safe Locking on Multiprogrammed Shared Memory Multiprocessors" (hereinafter "Mages"). Applicants traverse the rejection of claims 16, 33 and 50 for at least the reasons given above in regard to the claims from which they depend.

In regard to the rejections under both § 102(a) and § 103(a), Applicants assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time. Applicants reserve the right to present additional arguments.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/6000-33600/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: April 10, 2008